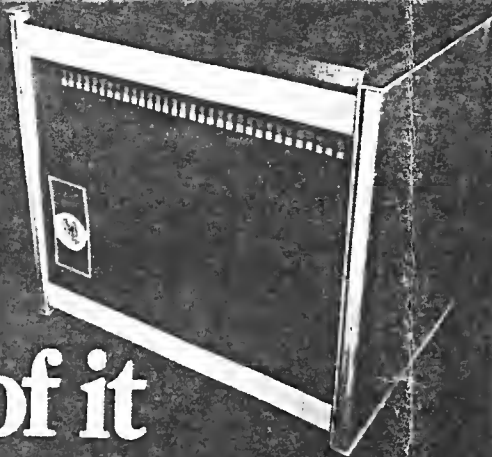


# The best time share multiplexer ever invented.

## And you can get rid of it in a year.



You shouldn't have to be stuck with a long lease. That's why you can rent our new Ultracom™ time share multiplexer for one year. (You can also rent it for up to five years or buy it outright. But isn't it nice to have a choice?)

And what a beauty you get.

You can transmit data from many terminals over a single voice grade line, saving thousands of toll calls a year. And accommodate up to 27 terminals over a 2400 baud line.

We speak in computer language, with assembled characters. Instead of unwieldy bit interleaving that some other people use.

You're sure of accurate transmissions because of our special parity check and unique synchronization.

And you reduce the number of lines you have to

lease, because your main data streams can go through intermediate stations, without interfering with transmission.

Finally, nobody else can match the service organization you get from Ultronic. Big enough to maintain over 10,000 terminals across the country today.

So, if you're thinking of cutting costs with time share multiplexing, why not get the equipment you can get rid of in a year?

Call (609) 235-7300. Data Communications Products Division, Ultronic Systems Corp., Mount Laurel Industrial Park, Moorestown, N.J. 08057.

**ULTRONIC SYSTEMS**  
**SYLVANIA**  
GENERAL TELEPHONE & ELECTRONICS

CIRCLE 33 ON READER CARD

Datamation

215 Hg

## IMMEDIATE ACCESS AND THE USER REVISITED

by Murray Turoff

In August of 1966, this author published in DATAMATION an article whose primary purpose was to alert the computer community to the dangerous situation confronting it from the then embryonic immediate access concept. In addition to that warning, an attempt was made to lay down fundamental guidelines for the most effective means of discouraging programmers from utilizing immediate access systems. Now, three years later, it seems appropriate to update those earlier considerations and evaluate the course of the battle as it now stands.

In general we can put ourselves on the back for the efficient and effective manner in which we have dealt with the situation. Our successes far outweigh the few failures that have occurred. The items we can be thankful for are basically the following:

The major computer manufacturers have been quite careful to build multipurpose executives which in principle will perform any possible function the computer may be asked to perform by any potential user. This executive concept of being all things to all people has put us in the beautiful situation of being able to deny to the immediate access user enough of the computer's power so that he is encouraged to go back to his slide rule or desk calculator.

The concept that every one must offer a FORTRAN and COBOL language on the system has been an exceedingly neat way of consuming manpower resources and preventing any deviates from spending their time on more dangerous concepts. In fact, the N language system concept (for which N is allowed to go to infinity) is one of the best possible approaches we have available to us. The larger the set of completely independent languages offered, the

greater ability we will have to make the user feel he can never master any significant part of the system.

One of our greatest achievements has been in the careful manner in which we have buried the JOSS philosophy. Except for some minor extensions for the purpose of on-line gaming, which seem to have been forced upon the computer people, RAND seems to have abandoned any extension of JOSS philosophy and has moved on to the bottomless pit of on-line graphics. We were worried when we heard a rumor that IBM was developing a JOSS-like system, and while we realize it is sometimes difficult to keep a tight rein on all the parts of an organization of that size, IBM as the industry leader must take special pains to insure it does not allow things like JOSS to emanate as an IBM-supported item. Apparently, the proper people found out about it,



Dr. Turoff is a member of the systems evaluation division, Office of Emergency Preparedness, in the Executive Office of the President. From 1964 to 1968 he was with the Institute for Defense Analysis. He has also been associated with the MIT computation center and IBM. He has a BA in physics and mathematics from the Univ. of California and a PhD in physics from Brandeis Univ.

The opinions expressed in this article are the responsibility of the author and do not necessarily reflect the views of his employer.

May 1969

stamp out  
nonprogrammer users

## IMMEDIATE ACCESS . . .

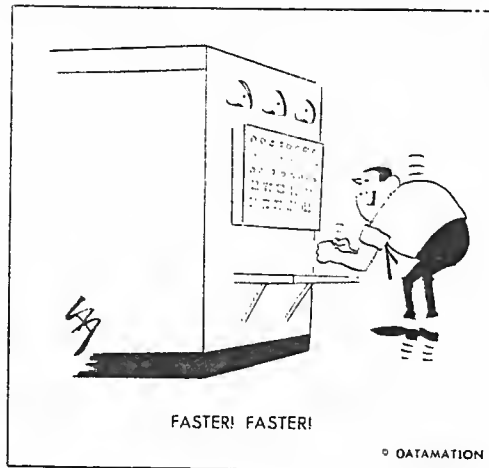
because current rumors are that if Ross does surface at all in IBM, it will only be allowed out as a Type 3 program and not as a system.

Bolt Beranek and Newman, one of those small companies that sometimes give us concern, did have the audacity to extend Ross to handle string manipulation. The very thought that a user might do his own string manipulation is an extremely dangerous concept. Fortunately, BB&N only developed that capability for a PDP-1, which really does not count at this point in time. Furthermore, they called their system SINCOMP and ISINCOMP, which alone is enough to discourage most users. Since most of the commercial Ross look-alikes do not attempt to extend the power of the system and even encapsulated it to some degree (removed the recursive string routine capability or the exact arithmetic) we can be more relaxed about the current situation.

If users inquire about Ross, be sure to inform them it is only a computerized desk calculator. Users must remain ignorant that Ross is more flexible than FORTRAN and that the concept can be extended to a master system control language for operating file, communication and symbol manipulation capability.

The only real danger we have had with respect to the Ross philosophy appears to have occurred at Massachusetts General Hospital. This has turned out to be an excellent example of the dangers involved in letting users gain control over a computer effort. The fact that such a system exists at all is a danger, and I suspect that our official view diplomatically should be that this is a very, very specialized application for a very, very specialized community of users to whom it is impossible to draw any general conclusions. Hopefully, you are point to the rising costs of medical care and blame it on the inefficient use of expensive computer hardware. As you all, we must insure they get as little publicity as possible.

It is extremely gratifying to note that most of the major manufacturers are going the Ross route and that their extended versions of BASIC are getting to look more and more like FORTRAN. Unfortunately, we cannot avoid satisfying the demands of those users interested in straightforward numerical tools. We must merely assure this capability is divorced and separated from any flexible and powerful control of data things. We must assure that if the user wishes data handling capability, he must come to the programmer and have a special short session answer system designed for his single application. It is also pleasing that BASIC systems remain noninterpretive so that users are forced to learn the



distinctions between composition, debugging and execution of a program as well as being forced to begin over for every little mistake.

The cooperation we have had from government in our effort is most gratifying. We can rest assured that with the exception of a few mavericks who have managed to discover a cut here or there in the red tape, most potential government users will never see an immediate access terminal they would consider using.

I must admit that for a while there was some concern that since the government had mistakenly financed a large portion of the initial R&D in the time-sharing, immediate access area that they would become a big user of such systems. Fortunately, there have been a couple of important factors which have managed almost completely to inhibit the government's movement in this direction. First, immediate access systems for users cost more than immediate access systems for programmers, which in turn cost more than batch-oriented systems, and of course batch systems always utilize the computer hardware more efficiently than immediate access systems. Second, we have carefully neglected to do a good job of analyzing the cost benefits that might be incurred from such indirect measures as user time saved, rapid availability of information, and opportunity costs. Therefore, the government examiner or auditor faced with nothing but intuitive pleas by those misfits trying to justify user-oriented systems must of course disallow the added expense.

Incidentally, we would like IBM to cut out those automatic typewriter commercials which illustrate how expensive the manual writing of a letter can be when properly costed out. We do not want budgetary minded individuals to infer from this there might also be indirect user costs associated with the writing of computer programs. In addition, we can be thankful for those rules which make it harder for the smaller companies to compete when they do not manufacture all the necessary peripheral equipment. The smaller companies are usually the mavericks outside the mainstream of the industry anyway. Finally, we must encourage the government to push harder for standards; now is the best time before users might get wise to other possible languages.

The situation to avoid at all costs is one in which the user is provided with a computer capability for improving the efficiency of his own functions in the user organization. This is typified by a simple-to-use but flexible personnel user file and communication system in which the user can store, edit or update his own data, memos or programs, and where he can send or receive parts of his or other user files. These fundamental operations are characteristic of the functions performed in almost any user community or organization and will spark the user's interest since it provides a day-to-day service to him. Then, of course, if one adds a computational and symbol manipulation capability that is coherent in language structure with the file and communication capability—one that can be learned in modules—the user will then be able to do a significant portion of his own work. We would in such an environment lose the priesthood aura that we have so carefully built.

### keep them down

After all, we have nurtured over the years the following precepts relating to the programmer-user interface:

1. The user is incapable of utilizing a computer by himself for meaningful work.
2. Only the programmer really knows what the user wants.
3. The user is incapable of stating his problem in a clear and concise manner compatible with the computer system.
4. All meaningful approaches to problem solution must by definition be compatible with a computer system.
5. The user must remain grateful and satisfied for the programmer's interpretation of his problem and the com-

puter solution to that interpretation.

An immediate access system with sufficient communication capability to allow the user to establish an interaction between him and the programmer on a common language foundation would be a severe blow at these commandments.

Even worse, such an approach might lead to the user community developing, over time, their own management information system able to adapt to the changes in the community structure and its environment. Consider what this would mean. All we have said about the effort and design study needed for management information systems and how these must follow the tried and true implementation procedures and techniques associated with information management systems.

Before the user becomes aware of the potential in flexible computer-assisted communications among users, terminals and files we should join hands with the telephone companies and ask government to come to our aid by disallowing such a capability as a fundamental violation of the principles of separation between computation and communication. I suspect that this will inhibit the use and development of computer assisted conferencing or on-line gaming is a small penalty to the society which we can quietly sweep under the rug. The community representing the soft sciences is, as a whole, still far enough behind in the use of computers that we would not expect any really organized opposition to such regulations. However, it is urgent that we seek adequate regulatory protection before these academic communities wake up.

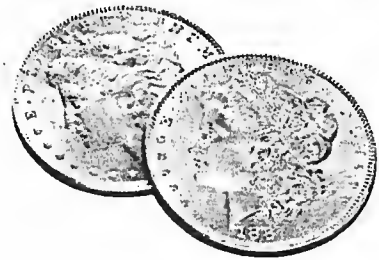
One dangerous area that has developed recently is the movement toward computer-assisted education. There are rumors of some of the efforts in this area are considering the possibility of developing user-oriented general purpose software which would allow teachers at a local school to develop their own educational programs tailored to their student community. I feel it would set a very dangerous precedent to treat teachers any different from other user communities. In addition, a few of the psychologists and educators becoming acquainted with computers are beginning to feel intuitively that there is some commonality between the foundations of their tools and techniques and the earlier efforts of information handling, data retrieval, and computerized systems.

Consequently, I urge the technique of fragmentation. Computer-assisted education is a very, very specialized application and has no relationship to other user-oriented systems. Separate languages and systems with highly educational-oriented programs and vocabularies must be built from machine language. There is no possibility that an immediate access language for the user could be used for both the construction of tailored data base retrieval applications as well as tailored educational response systems.

Be sure to isolate the development group for management information systems from the group working on educational packages (separate cities is desirable) as well as the group concerned with on-line computation. Cross-fertilization is the fundamental danger to be avoided at all costs. Adequate justification, administratively, may be found by illustrating the degree of departmentalization existing at most of the universities and by pointing to the limited technical communication existing among academic departments and how well their example highlights the lack of meaningful computer research or development problems requiring company interdivisional cooperation.

At a community, we have had a rewarding three years; however, the same seed that existed three years ago still exists and only by constant vigilance and weeding will our community continue to blossom and flourish. We can foresee, if we maintain our current direction, a glorious future in which the demand for members of the computer community will eventually exceed the population.

## A COMPLETE IN-HOUSE TIME SHARED SYSTEM FOR UNDER \$2.00 PER TERMINAL HOUR?



### Swinging!

- A typical WESTIME 300 SYSTEM consists of a 16K IBM 1130 with a 15 channel WTI transmission controller.
- Languages can be either CONVERSATIONAL FORTRAN or BASIC.
- Terminals can be IBM 2741 or 33 teleprinter (mixed if you like).

FOR COMPLETE DETAILS  
CALL OR WRITE TODAY.

**wti** western telematic inc.  
5507 Peck Road, Arcadia, California 91006 • (213) 442-1862  
CIRCLE 34 ON READER CARD